

Proceedings of the 59th CIRP Conference on Manufacturing Systems (CMS) 2026

Adversarial Agentic LLM based optimization for factory layout and dispatching selection

Johannes Pflüger^a, Wei Ye^b, Xingyu Li^b, Sang-Gook Kim^c, Marvin Carl May^{c,d,1}^aDepartment of Industrial Engineering, Karlsruhe Institute of Technology (KIT), Kaiserstr. 12, 76131 Karlsruhe, Germany^bSchool of Engineering Technology, Purdue University, West Lafayette, IN, United States^cDepartment of Mechanical Engineering, Massachusetts Institute of Technology (MIT), 77 Massachusetts Avenue, Cambridge, MA 02139, USA^dSchool of Mechanical and Aerospace Engineering, Nanyang Technological University (NTU), 50 Nanyang Avenue, 369798 Singapore, Singapore* Corresponding author: Marvin Carl May, E-mail address: marvin.may@ntu.edu.sg

Abstract

Heuristic methods remain vital for production scheduling and dispatching, yet manual design limits adaptability. This work introduces the Adversarial Agent-Based Large Language Model Evolutionary Algorithm (LLaMEA) Framework, integrating Large Language Models (LLMs) into evolutionary optimization to autonomously generate and refine heuristic strategies. A dual-agent workflow proposes and evaluates production strategies through iterative improvement. Evolutionary operators combined with LLM-driven reasoning to evolve interpretable solutions in simulation-based environments. A persistent archive captures successful solutions across optimization runs, enabling knowledge transfer and accelerating convergence in new tasks. This knowledge-driven mechanism distinguishes the approach from purely generative methods through contextual adaptation of accumulated expertise. Validation on a matrix production benchmark confirms LLM-generated strategies achieve Pareto-optimal performance comparable to genetic algorithms while maintaining transparency and requiring significantly less manual effort, establishing a scalable foundation for adaptive production optimization.

© 2026 The Authors. Published by Elsevier B.V.

This is an open access article under the CC BY-NC-ND license (<https://creativecommons.org/licenses/by-nc-nd/4.0>)

Peer-review under responsibility of the scientific committee of the 2025 CIRP CMS

Keywords: Large Language Models; Production Planning; Knowledge Management

1. Introduction

Production planning and control in matrix manufacturing systems pose particular challenges. Unlike traditional, linear production lines, matrix based systems offer a flexible, modular structure where products can follow multiple routes through shared resources [1]. This flexibility leads to complex optimization problems as layout decisions, transportation rules, and machine sequencing must be coordinated simultaneously across several often conflicting objectives [2].

The consistent application of heuristics has proven effective for handling optimization tasks in practice, especially when fast and sufficiently good solutions are more important than exact but computationally expensive optima [3]. Widely known scheduling rules such as “Shortest Processing Time” (SPT) or “First-In-First-Out” (FIFO) highlight this [4]. However, the adaptability of these manual methods is limited: changes in op-

erational conditions within the matrix production system often require time-intensive redesign and experimentation [5]. Similarly, optimality is not guaranteed. Thus, genetic algorithms are commonly used to optimize layout [6] or create novel heuristics [7] but are difficult to interpret. This reduces acceptance among production managers, as the rationale behind certain solution proposals is not transparent [8, 5]. Solution quality also hinges on model formulation skills [7].

Meanwhile, Large Language Models (LLMs) open new possibilities to interpret text [9] but also for automatically generating, interpreting, and continuously improving heuristic approaches [10]. LLMs can work with complex problem descriptions in natural language and generate structured solution concepts, including code and heuristic strategies [11]. Despite promising capabilities, challenges remain such as inconsistent outputs, plausible but incorrect solutions, and high computational demands [12]. The key question in this paper is how to effectively integrate LLMs into systematic optimization frameworks, particularly by combining them with evolutionary mechanisms, to produce competitive and comprehensible solutions

for manufacturing-specific problems. Matrix production systems with their combinatorial decisions and diverse constraints provide an ideal application domain, as they demand high flexibility, adaptability, and transparency [1, 13].

This work targets three central limitations inherent to current optimization approaches. First, it mitigates the need for labor-intensive manual heuristic design and enhances adaptability to changing operational conditions. Second, it lowers the knowledge barrier typically associated with traditional evolutionary algorithms, thereby improving accessibility for practitioners. Third, it addresses the insufficient validation of LLM-based optimization methods within realistic industrial production settings.

This work is structured as follows. Section 2 describes the recent research regarding the generation of heuristics and LLM applications in manufacturing. Then, Section 3 describes the proposed framework. Subsequently, Section 4 applies the proposed method to the matrix production problem as a case study. The work finishes with a conclusion in Section 5.

2. State-of-the-art

This section presents a concise overview of the state of the art in two main areas relevant to the proposed approach.

2.1. Evolutionary Programming and Genetic Approaches

Multi-objective optimization has advanced significantly with evolutionary algorithms designed to tackle complex, real-world challenges [7]. Classical methods such as NSGA-II and NSGA-III utilize Pareto dominance and niching mechanisms to maintain diversity, while decomposition-based approaches like MOEA/D and RVEA enable problem partitioning and guided search via reference directions [14]. Conventional metaheuristics including simulated annealing and tabu search remain valuable for convergence-critical tasks. Despite algorithmic maturity, bridging the gap between benchmark success and practical industrial applicability remains challenging due to the complexity of constrained, stochastic, and irregular Pareto landscapes [14]. [6] demonstrate an integrated multi-objective framework jointly optimizing layout and production control in manufacturing, achieving superior results over sequential methods using Non-dominated Sorting Genetic Algorithm II (NSGA-II) combined with discrete-event simulation. [15] comparatively assessed NSGA-II, simulated annealing, and tabu search for reconfigurable manufacturing, revealing trade-offs in convergence speed and solution quality under identical time constraints.

2.2. Large Language Model-Based Optimization Approaches

Recent breakthroughs in LLMs reshape algorithm design by enabling automatic generation, selection and evolutionary refinement of optimization strategies [16]. Huang et al.'s ORLM framework fine-tunes open-source LLMs on extensive industrial datasets to translate natural language problems into executable mathematical models, but it oversimplifies complex

scenarios [17]. The LLaMEA system advances this by integrating LLMs in an evolutionary (1+1) loop to generate, mutate, and select Python-based optimization algorithms guided by benchmark feedback, incorporating error-informed self-debugging; its LLaMEA-HPO (Hyperparameter Optimization) variant further offloads hyperparameter tuning to SMAC3, cutting query requirements by 95% without loss in quality [18, 19]. The Evolution of Heuristics (EoH) framework merges natural language and code representations in population-based evolution, applying multi-view genetic operators simultaneously [20]. ReEvo leverages reflective hyper-heuristics that utilize verbal gradient-like signals to guide heuristic crossover [21]. Additionally, LLM-powered power scheduling systems emphasize translating user intent into formal optimization tasks through multi-agent architectures instead of direct algorithm generation [22]. Collectively, these LLM-based methods illustrate complementary paradigms—ranging from domain-specific fine-tuning [23] and prompt-based generation [24] to hybrid evolutionary - code evolution and reflective guidance — each addressing distinct facets of automated heuristic design and problem specification [22].

The integration of advanced optimization into production planning is significantly challenged by the increasing complexity and dynamism of industrial environments. Current LLM-based approaches and evolutionary algorithms often lack the scalability, transparency, and practical adaptability required to meet real-world demands [25]. Recent research on Generative Manufacturing Systems addresses these issues by employing generative artificial intelligence to orchestrate heterogeneous autonomous manufacturing resources under human supervision, thereby enabling flexible decision-making, enhanced system resilience, and enriched human-autonomy collaboration [26].

3. Agent-Based LLaMEA

The Agent-Based LLaMEA system presents a novel framework for production optimization that integrates artificial intelligence, evolutionary algorithms, and linguistic reasoning. It replaces purely numerical optimization with language-based analytical and generative reasoning, allowing decision-making processes to capture human-understandable strategic dimensions rather than relying solely on computational outputs. The system uses LLM (agents) as both strategy creators and evaluators within an iterative evolutionary loop, generating, critiquing, and refining production strategies across generations until the termination criteria is reached, as visualized in Figure 1.

While the framework is generalizable, we restrict its application to manufacturing, based on the industrial use case reported by [6], visualized in terms of decision making flexibility in Figure 2. Specifically, both the layout and material flow rules have to be holistically optimized. Eight machines have to be positioned on the shopfloor, both with predetermined possible locations or freely with respect to constraints. Secondly, the dispatching rules for material flow and the scheduling rules for selecting the next job at a machine can each follow First-

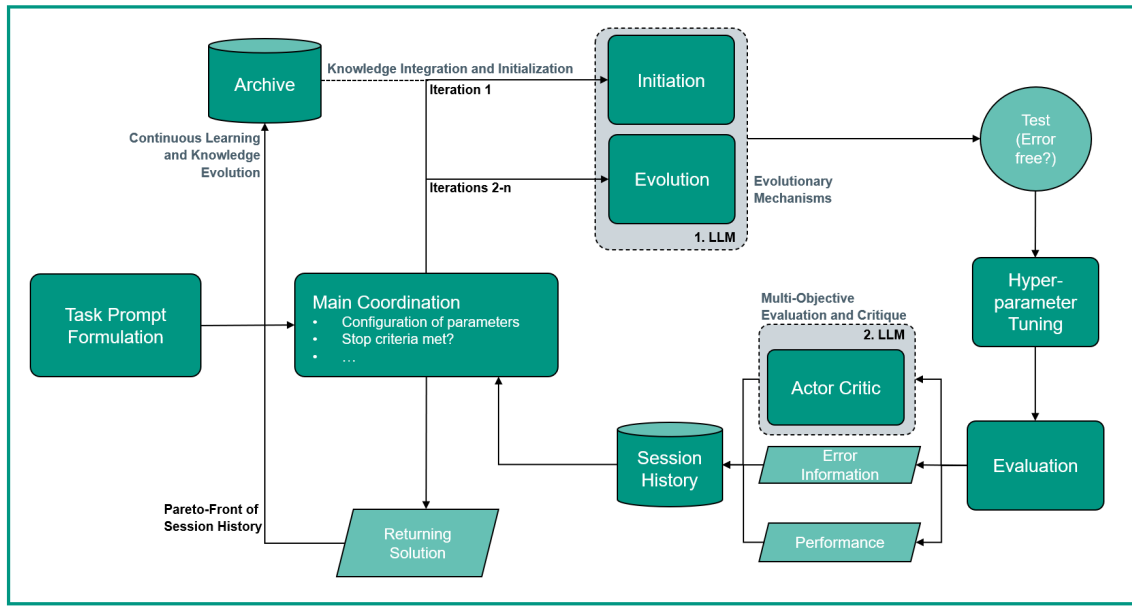


Fig. 1. Agent-Based LLaMEA Framework

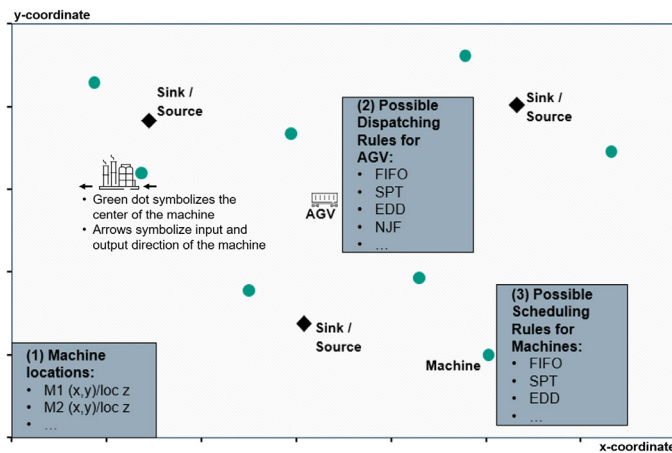


Fig. 2. Exemplary problem scope based on [6]

In-First-Out (FIFO), Shortest-Processing-Time (SPT), Earliest-Due-Date (EDD) among others.

3.1. Knowledge Integration and Initialization

At the beginning of each optimization run, Agent-Based LLaMEA starts by integrating knowledge from previous optimization sessions to bootstrap among different problems based on a continuous learning and history and archive filled at the end of a successful optimization. The system accesses an archive of successful strategies by comparing problem descriptions of previous and the problem at hand to initialize its population with proven approaches. This ensures cumulative learning across different problem scenarios and may as well lead to novel approaches. Two specialized agents, an Actor and a Critic, are instantiated. The Actor proposes new strategies, while the Critic assesses and refines them through structured analysis following a

PRESERVE/IMPROVE/DIRECTION schema. This adversarial dual-agent structure balances exploration with exploitation. The initial population generation combines direct reuse of historical solutions with newly constructed strategies created from diverse domain heuristics and expert-driven input. This hybrid initialization approach promotes diversity in the search space while ensuring strategic consistency across the population.

3.2. Evolutionary Mechanisms

The optimization process unfolds through iterative evolutionary cycles where the best-performing strategies act as parents for subsequent generations. The framework uses two primary genetic operations: mutation and crossover. Mutation alters individual strategies to target specific improvements such as throughput enhancement, quality increase, or resource efficiency. Crossover in our case, by contrast, merges a good parent with a random one to integrate promising elements from the second one into the first. This produces hybrid offspring that can inherit complementary strengths from both. Randomization helps balancing exploitation with exploitation.

An adaptive and to one's needs configurable temperature annealing mechanism regulates the creativity of LLM-generated proposals, starting with high-temperature exploration in early phases and gradually decreasing temperature for focused optimization as convergence approaches. Since there are two adversarial LLMs used in this approach the temperature for generation and critiquing can be set independently.

3.3. Multi-Objective Evaluation and Critique

Each candidate strategy undergoes multi-layer evaluation, starting with error testing. An evaluation function, in this case a production system simulation, assesses quantitative performance indicators such as throughput, delay, lead time, and re-

source utilization. Afterwards, the adversarial Critic agent provides qualitative analysis to highlight strengths, weaknesses, and improvement directions. This linguistic feedback is reintegrated into the next generation's LLM prompts, guiding learning in a semantically rich and context-aware manner and making the prompt for mutation or crossover unique for each solution. A central feature of Agent-Based LLaMEA is how it copes with the Pareto front for multi-objective optimization. Rather than collapsing competing goals into a single aggregated metric, the system identifies and preserves all non-dominated solutions, enabling decision-makers to select context-relevant strategies aligned with varying production priorities. However, there is an aggregated fitness value in use for choosing part of the parents for the next evolution, similarly to NSGA-II.

3.4. Modularity and overall approach

First, each agent with prompts ϕ and input γ can be understood as a function $a_i : \phi_{system} \times \phi_{task} \times \gamma_{i,t} \rightarrow \omega_{i,t}$. Agent 1, the actor's role is that of ϕ_{system} being creator, tasked with creating a feasible, good solution (ϕ_{task}), given the former solutions and critic input $\gamma_{i,t}$. Its solution $\omega_{i,t}$ in this case is a tuple of machine positions, dispatching and scheduling rules to be used. The second agent, the critic, similarly takes these solutions $\omega_{i,t}$ and the discrete event simulation performance $DES(\omega_{i,t})$ as input and produces a constructive critic for each solution. Second, once iterative improvements converged, the problem definition (part of $\phi_{system} \times \phi_{task}$), final solution $\omega_{1,end}$ and critique $\omega_{2,end}$ are retained as persistent knowledge and become accessible to the first agent for initial population generation. Third, making this persistent knowledge base and the session history available in $\gamma_{1,t}$ supports reuse and population upgrade, as temperature annealing (mainly for a_1) controls creativity. Fourth, a state-graph oriented orchestration, herein implemented as a Lang-Graph state machine, using Google's Gemini 2.5 flash, governs the optimization loop. Last but not least, customization of prompts, even to different problems, the evaluation, potentially beyond simulative evaluation, and overall workflow adaptability ensure widespread applicability in manufacturing and across optimization problems and complex decision-making contexts.

4. Case study

The matrix production system investigated in this study developed by [6] making the solutions comparable. It represents a reconfigurable manufacturing environment designed to replace rigid, linear production lines with a flexible, modular layout. Operating within a factory space of fixed dimensions, the system comprises strategically placed input stations (sources), eight processing machines grouped by capability, output stations (sinks), and a single automated guided vehicle (AGV) responsible for material transportation. The problem centers on discovering optimal combinations of machine (p_i) and heuristic dispatching rules for both transportation (t) and processing (m) to balance four, often conflicting, objectives: throughput maximization (f_t), delay minimization (f_d), utilization maximiza-

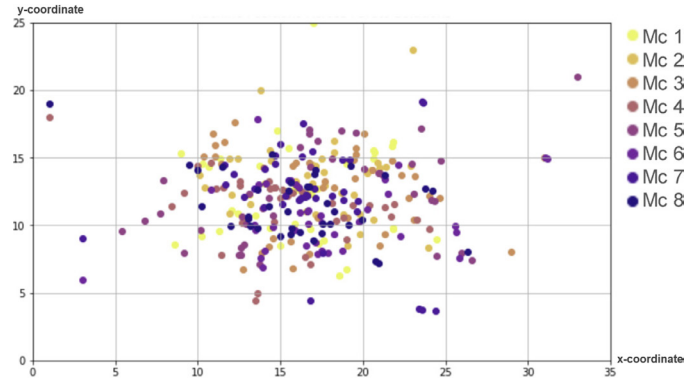


Fig. 3. Machine Positions on the 25x35-Grid Across Pareto Solutions from [6]

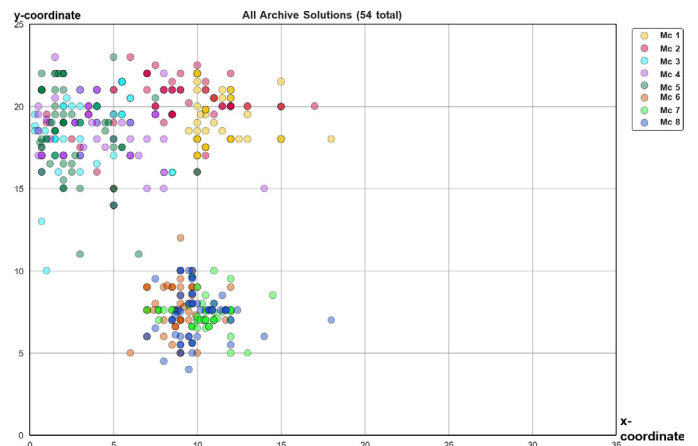


Fig. 4. Layouts of all archive solutions of the method on the shopfloor

tion (f_u), and lead time reduction (f_l). The complexity of the problem arises from the mix of discrete decisions, such as dispatching policies, compounded by stochastic disturbances like machine failures and processing time variability. This necessitates a robust multi-objective optimization approach capable of navigating a large search space under real-world industrial constraints and dynamic behavior. All in all: A solution consists of eight coordinates representing the placement of the machines and two rules, one transport-rule determining in which order the AGV handles the jobs and one machine rule determining in which order the machines handle the jobs in their buffer.

The evaluation utilized a simulation-based optimization framework to compare the performance of the agent-based LLM-driven approach to the widely used NSGA-II. By applying identical system parameters and decision frameworks, the study ensured methodological compatibility for an unbiased performance comparison. The LLM-based system generated Pareto-optimal production configurations by exploring the solution space with the aforementioned techniques. These mechanisms allowed the LLM approach to systematically produce diverse and high-quality solutions without extensive domain-specific programming effort.

4.1. Original Use-Case

Agent-Based LLaMEA is evaluated on two variations of the problem: a more constrained version, with pre-determined machine placement locations (the real industrial use-case) and then relaxed to the matrix like grid (with industrial constraints), as well as a more extensive version, involving a larger number of machines, an expanded shop floor, and additional sources and sinks to consider. In both scenarios, the proposed program successfully generated valid solutions and demonstrated consistent reasoning across the distinct problem instances. However, in the first case, the performance of Agent-Based LLaMEA did not surpass that of the genetic algorithm as one can see in comparing the results shown in Tables 1 and 2. This outcome may be attributed to the relatively small size of the, now almost permutation-like, problem, which can lead to repetitive reasoning patterns and consequently similar solutions.

Table 1. Solutions Selected from Archive Pareto Front

t, m	p_i	f_t	f_d	f_u	f_i
NJF, EDD	[1, 2, 3, 4, 5, 8, 7, 6]	0.322	21.3	0.823	99.8
EDD, EDD	[2, 3, 4, 5, 1, 8, 7, 6]	0.302	17.6	0.748	98.7
NJF, SPT	[5, 4, 1, 3, 2, 6, 7, 8]	0.310	26.1	0.765	91.9
EDD, EDD	[2, 3, 1, 4, 5, 8, 7, 6]	0.321	18.8	0.753	95.8

Table 2. Solutions from the output of the genetic algorithm (based on [6])

t, m	p_i	f_t	f_d	f_u	f_i
NJF, SPT	[1, 3, 5, 4, 6, 7, 8, 2]	0.338	27.1	0.840	99.6
FIFO, SPT	[5, 2, 1, 4, 7, 8, 6, 3]	0.325	23.6	0.819	95.6
EDD, EDD	[2, 8, 1, 3, 4, 6, 5, 7]	0.326	18.2	0.777	95.8
EDD, FIFO	[3, 8, 4, 6, 5, 7, 2, 1]	0.327	18.8	0.811	91.8

In the extended version the program shows no signs of problems handling the task. Since it is specially designed, there are no alternative solutions, so it is not possible to compare the quality of the generated solutions. However, the results show that the program is capable of handling more complex and scalable tasks.

4.2. Main Use-Case

The main use-case, described above, had no predetermined machine locations and comparative analysis of solutions highlights that the LLM-based framework achieved solution quality on par with the state-of-the-art genetic algorithm (to be seen in Table 3 and 4), even surpassing the genetic algorithm in some cases. Notably, as one can see in the Figures 3 and 4, our LLM based approach demonstrates a distinct reasoning pattern by clustering machines near their associated sources and sinks to prioritize transport efficiency, contrasting with the genetic algorithm's more varied spatial distributions. The trade-off relationships between objectives, for example, the increase of throughput leading to larger lead times and delays, were similarly captured by both approaches, confirming the approach's ability to model the underlying problem dynamics. Importantly, while crossover operations played a rather minimal role in this study's results, mutation and the archiving of past solutions are critical to sustaining search effectiveness and diversity as visible over temporal improvement. This does not have to mean

that the crossover mechanism is not useful for any problem instance. In other instances with a less defined solution space it may profit greatly from this mechanism combining different approaches.

Table 3. Solutions Selected from Archive Pareto Front

t, m	p_i	f_t	f_d	f_u	f_i
EDD, EDD	[10, 22] [8, 19] [0.7, 17] [3, 19] [0.7, 21] [8.7, 7.6] [10.7, 7.6] [9.7, 9]	0.32	16	0.78	94
NJF, SPT	[15, 22] [13, 19] [3, 18] [6, 20] [9, 17] [12, 7] [7, 8.5] [10.5, 4]	0.33	27	0.82	91
NJF, EDD	[10, 22] [8, 19] [0.7, 16] [3, 19] [0.7, 22] [7.7, 7.6] [11.7, 7.6] [9.7, 9.6]	0.34	20	0.83	99
NJF, EDD	[10, 18] [12, 18] [2, 17] [4, 17] [4, 19] [7, 11] [9, 11] [8, 9]	0.34	20	0.82	95

Table 4. Solutions from the output of the genetic algorithm (based on [6])

t, m	p_i	f_t	f_d	f_u	f_i
SPT, SPT	[12, 11] [21, 14] [17, 15] [16, 8] [24, 12] [26, 9] [24, 4] [24, 13]	0.34	26	0.84	92
EDD, EDD	[23, 15] [11, 14] [13, 18] [15, 13] [12, 12] [18, 11] [15, 14] [18, 10]	0.32	17	0.77	94
FIFO, FIFO	[22, 10] [20, 14] [18, 13] [14, 5] [26, 8] [31, 15] [24, 19] [21, 7]	0.33	22	0.76	90
NJF, FIFO	[17, 25] [23, 23] [31, 15] [1, 18] [33, 21] [3, 6] [3, 9] [1, 19]	0.34	22	0.82	93

Overall, the results in both sub-use cases validate the potential of the Agent-Based LLaMEA framework as a versatile and scalable alternative to classical evolutionary algorithms for complex industrial production optimization problems. The framework shows promise not only in solution quality but also in reducing the need for intricate algorithm design and domain expertise. This study thus demonstrates that natural language-based LLM queries can effectively substitute traditional optimization algorithm implementations without significant loss in performance, opening avenues for more accessible and adaptive manufacturing system optimization.

5. Conclusion

The Adversarial Agent-Based LLaMEA framework introduces a novel approach to strategic production optimization by leveraging LLMs within a dual-agent system that separates creative exploration from analytic evaluation. This design enables cumulative learning and the integration of cross-domain knowledge while producing natural language strategies that are accessible to practitioners without specialized optimization training. The framework achieves comparable performance to established genetic algorithms on benchmark tasks, demonstrating its potential as a tool for bridging expertise gaps.

However, the approach is computationally intensive relying in LLM inference and requires accurate simulation environments, which restricts its real-time applicability. Furthermore, dependence on external APIs and the opaque nature of LLM reasoning raise concerns about scalability and reliability. These factors currently position Agent-Based LLaMEA as a primarily strategic planning tool rather than an operational solution. Despite this heavy LLM focus and integration into the optimization loop, in principle, the agents could be implemented by humans or other means to ensure domain knowledge integration.

Still, the most feasible implementation will focus on LLMs and other computational natural language reasoning approaches.

Future advances should focus on enhancing computational efficiency through parallelization and the use of domain-specific, locally deployed LLM models. Methodological developments such as continuous learning, adding agents and/or agentizing the framework by giving these agents the power to orchestrate the tool usage should be regarded. This step is already partly prepared by integrating of the LangGraph framework. Beyond manufacturing, the framework holds promise for applications in dynamic scheduling and dispatching, logistics, service operations, and urban systems, where multi-objective optimization and transparent decision-making are critical.

In a nutshell, Agent-Based LLaMEA represents a promising step towards making advanced optimization methods more accessible by combining interpretability with less modelling effort and, thus, less needed modeling expertise. Continued research is needed to improve robustness, scalability, and trustworthiness to realize its full potential cross-domain contexts.

Acknowledgements

This research was partially funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) "L2M4PC" under Project ID MA 11729/1. The authors used ChatGPT (OpenAI) and Claude (Anthropic) only to improve readability and language, and then reviewed/edited the text and take full responsibility for the content. No AI tools were used for problem formulation, analysis, or deriving research insights.

References

- [1] D. Ranke, M. Trierweiler, P. Berkhan, M. Münnich, T. Bauernhansl, Definition and Description of Matrix Production Systems, *Procedia CIRP* 130 (2024) 1875–1880. doi:10.1016/j.procir.2024.10.331.
- [2] P. Greschke, M. Schönmann, S. Thiede, C. Herrmann, Matrix Structures for High Volumes and Flexibility in Production Systems, *Procedia CIRP* 17 (2014) 160–165. doi:10.1016/j.procir.2014.02.040.
- [3] M. C. May, L. Schäfer, A. Frey, C. Krahe, G. Lanza, Towards product-production-codesign for the production of the future, *Procedia CIRP* 119 (2023) 944–949. doi:10.1016/j.procir.2023.02.172.
- [4] M. C. May, L. Overbeck, M. Wurster, A. Kuhnle, G. Lanza, Foresighted digital twin for situational agent selection in production control, *Procedia CIRP* 99 (2021) 27–32. doi:10.1016/j.procir.2021.03.005.
- [5] J. Branke, S. Nguyen, C. W. Pickardt, M. Zhang, Automated Design of Production Scheduling Heuristics: A Review, *IEEE Transactions on Evolutionary Computation* 20 (1) (2016) 110–124. doi:10.1109/TEVC.2015.2429314.
- [6] W. Ye, J. Dvorak, X. Li, M. C. May, Dynamic integrated simulative layout planning and production control for matrix production in a semiconductor environment, *Procedia CIRP* 130 (2024) 1695–1700. doi:10.1016/j.procir.2024.10.302.
- [7] S. Shady, T. Kaihara, N. Fujii, D. Kokuryo, A novel feature selection for evolving compact dispatching rules using genetic programming for dynamic job shop scheduling, *International Journal of Production Research* 60 (13) (2022) 4025–4048.
- [8] M. Kumar, M. Husain, N. Upreti, D. Gupta, Genetic Algorithm: Review and Application, *SSRN Electronic Journal* (2010). doi:10.2139/ssrn.3529843.
- [9] K. Theuner, T. M. Elmgren, A. Götling, M. C. May, H. Akay, Weaving knowledge graphs and large language models (LLMs): Leveraging semantics for contextualized design knowledge retrieval, *Procedia CIRP* 134 (2025) 1125–1130. doi:10.1016/j.procir.2025.03.073.
- [10] M. C. May, T. Kaihara, Large language models for heuristics to solve aggregate production planning, in: H. Mizuyama, E. Morinaga, T. Nonaka, T. Kaihara, G. von Cieminski, D. Romero (Eds.), *Advances in Production Management Systems. Cyber-Physical-Human Production Systems: Human-AI Collaboration and Beyond*, Springer Nature Switzerland, Cham, 2026, pp. 387–400. doi:10.1007/978-3-032-03538-7_28.
- [11] H. Naveed, A. U. Khan, S. Qiu, M. Saqib, S. Anwar, M. Usman, N. Akhtar, N. Barnes, A. Mian, A Comprehensive Overview of Large Language Models (Oct. 2024). doi:10.48550/arXiv.2307.06435.
- [12] R. Patil, V. Gudivada, A Review of Current Trends, Techniques, and Challenges in Large Language Models (LLMs), *Applied Sciences* 14 (5) (2024) 2074. doi:10.3390/app14052074.
- [13] S. Kiener, N. Maier-Scheubeck, R. Obermaier, M. Weiß, D. G. Oldenbourg, *Produktionsmanagement*, de Gruyter Studium, De Gruyter Oldenbourg, 2018.
- [14] J. Zatarain Salazar, D. Hadka, P. Reed, H. Seada, K. Deb, Diagnostic benchmarking of many-objective evolutionary algorithms for real-world problems, *Engineering Optimization* 57 (1) (2025) 287–308. doi:10.1080/0305215X.2024.2381818.
- [15] S. Behrendt, M. Wurster, M. C. May, G. Lanza, Extended Production Planning of Reconfigurable Manufacturing Systems by Means of Simulation-based Optimization, *CPSLPublisher: Hannover : publish-Ing.* (2023). doi:10.15488/13440.
- [16] Y. Tan, B. Wang, W. Yuan, Q. Zhou, W. Ye, M. C. May, X. Li, S.-G. Kim, R. Athinayanan, Generative ai-driven disruption management in matrix production: A digital twin-based framework for resilient operations towards industry 5.0, in: *ASME International Mechanical Engineering Congress and Exposition, Vol. IMECE2025*, American Society of Mechanical Engineers, 2025, p. 167119. doi:10.1115/IMECE2025-167119.
- [17] C. Huang, Z. Tang, S. Hu, R. Jiang, X. Zheng, D. Ge, B. Wang, Z. Wang, ORLM: A Customizable Framework in Training Large Models for Automated Optimization Modeling, *Operations Research* (2025) opre.2024.1233ArXiv:2405.17743 [cs]. doi:10.1287/opre.2024.1233.
- [18] N. v. Stein, T. Bäck, LLaMEA: A Large Language Model Evolutionary Algorithm for Automatically Generating Metaheuristics (Jan. 2025). doi:10.48550/arXiv.2405.20132.
- [19] N. Van Stein, D. Vermetten, T. Bäck, In-the-loop Hyper-Parameter Optimization for LLM-Based Automated Design of Heuristics, *ACM Transactions on Evolutionary Learning and Optimization* (2025) 3731567doi:10.1145/3731567.
- [20] F. Liu, X. Tong, M. Yuan, X. Lin, F. Luo, Z. Wang, Z. Lu, Q. Zhang, Evolution of Heuristics: Towards Efficient Automatic Algorithm Design Using Large Language Model (Jun. 2024). doi:10.48550/arXiv.2401.02051.
- [21] H. Ye, J. Wang, Z. Cao, F. Berto, C. Hua, H. Kim, J. Park, G. Song, ReEvo: Large Language Models as Hyper-Heuristics with Reflective Evolution (Oct. 2024). doi:10.48550/arXiv.2402.01145.
- [22] T. Mongaillard, S. Lasaulce, O. Hicheur, C. Zhang, L. Bariah, V. S. Varma, H. Zou, Q. Zhao, M. Debbah, Large Language Models for Power Scheduling: A User-Centric Approach, arXiv:2407.00476 [cs] (Nov. 2024). doi:10.48550/arXiv.2407.00476.
- [23] T. Körner, M. C. May, M. Huber, LLMs as data preprocessors for natural language processing in manufacturing, in: *2025 IEEE 30th International Conference on Emerging Technologies and Factory Automation (ETFA)*, IEEE, 2025, pp. 1–7. doi:10.1109/ETFA65518.2025.11205804.
- [24] M. C. May, P. Liang, S.-G. Kim, Agentic llm-based contingency management in production control, *Procedia CIRP* 139 (2026) 373–378. doi:10.1016/j.procir.2025.08.195.
- [25] M. Korth, M. Wurster, M. C. May, G. Lanza, Framework for natural language processing to automate material flow simulation in production system planning, *Procedia CIRP* 138 (2026) 174–179. doi:10.1016/j.procir.2026.01.031.
- [26] X. Li, A. Nassehi, H. Yang, F. Tao, J. W. Sutherland, L. Wang, R. X. Gao, Generative manufacturing systems, arXiv (2026). doi:arXiv:submit/6711968.